|  |
| --- |
| **Produce User Documentation** |
| Writing Effective Documentation |
|  |
| [Type the abstract of the document here. The abstract is typically a short summary of the contents of the document. Type the abstract of the document here. The abstract is typically a short summary of the contents of the document.] |
|  |
| **Joanne McEvoy** |
| **9/10/2008** |
|  |

Contents

[Produce User Documentation 3](#_Toc208848137)

[Understand the system 3](#_Toc208848138)

[Gaining Knowledge of the system 3](#_Toc208848139)

[Examine existing documentation 3](#_Toc208848140)

[Original system specifications 3](#_Toc208848141)

[Design specifications 4](#_Toc208848142)

[Technical specifications 4](#_Toc208848143)

[Existing user documentation 4](#_Toc208848144)

[Activity 2.1 5](#_Toc208848145)

[Using the system to gain knowledge 6](#_Toc208848146)

[Activity 2.2 7](#_Toc208848147)

[Analysing the system 7](#_Toc208848148)

[Using a systemic approach 7](#_Toc208848149)

[Activity 2.3 8](#_Toc208848150)

[Writing Effective User Documentation 9](#_Toc208848151)

[Writing user documentation 9](#_Toc208848152)

[Know what you want to say before you say it! 9](#_Toc208848153)

[Know who you are saying it to! 9](#_Toc208848154)

[Know how to say it! 10](#_Toc208848155)

[Writing in plain English 10](#_Toc208848156)

[Making it clear 10](#_Toc208848157)

[Making it concise 11](#_Toc208848158)

[Making it correct 11](#_Toc208848159)

[Activity 2.4 11](#_Toc208848160)

[Sequencing the documentation 12](#_Toc208848161)

[Activity 2.5 13](#_Toc208848162)

# Produce User Documentation

## Understand the system

Before you start to create your user documentation it is important to get a comprehensive understanding of the system you are documenting. This can be achieved by:

* Looking at the system and answering the questions: “What does it do, and how does it work?”
* Looking at the original specifications of the system to find out what it has been designed to do.

The second task is then to begin writing the required documentation. This may present the most challenging part of the process, so you will learn how to write effectively to your audience – the users.

## Gaining Knowledge of the system

Two main ways of learning about a system are to:

* Examine any existing documentation (such as technical documentation and/or specifications)
* Use the system to find out how it works.

## Examine existing documentation

All system should have basic documentation available, although basic systems may combine or compress some of the standard documentation. For example, original design specifications and system specifications are sometimes combined in small systems.

There are a number of resources you can use to find out about a system you are about to document. Some of the more important ones include:

* Original system specifications
* Design specifications
* Technical specifications
* Existing user documentation

## Original system specifications

The original specifications, especially the user requirements, explain the function of the system. The high-level description of the system’s functions can often provide you with information about the functions you need to document. Because specifications describe who the system is for and how it will be used, they will also give you an idea of what sort of people will be using the system and when and how they will use is. This is a big help in making sure that your documentation is relevant to your audience.

System specifications are usually not too technical, as they are to be understood by both the technical and non-technical people. When you are documenting a system, the most useful part of the original system specification is often the user requirements description. This should be a fairly short explanation of what the user wants and expects from the system.

## Design specifications

Design specifications are prepared to provide a complete picture of what the system will look like and how it will operate.

The role of design specifications makes a valuable source of information when you are documenting the system. For example, a typical design specification will show illustrations of screen designs and explain how a user will move form one screen to another. The illustrations make the content easy to understand and it becomes simple to match the plan to what is actually on the screen. The design specifications are also a good place to find out how the system should work (for example, what the function keys and menus do).

## Technical specifications

Technical specifications describe the technical features of the system. For example, technical specifications or a database application might describe the tables and the fields of data that are to be collected. This would include data such as the size of a particular field. Most technical specifications describe features such as:

* Expected response times
* System requirements (the minimum required software and hardware configuration)
* What computer and operating system environment the system is designed to operate with.

If you use technical documentation as past of the user documentation, remember that it is not written for the general audience. You may need to ass explanations, explain issues differently or in other ways adapt technical information for non-technical audiences.

## Existing user documentation

In some case there may already be a user documentation associated with a system, and this can be used to help you learn the system you are documenting. For example, a user may have already prepared a basic user manual for other users, or the system may include help features, such as a search feature or step-by-step help topics.

### Activity 2.1

1. Find examples of each type of documentation discussed, and review each type to identify the differences between each type of documentation and the intended audience (reader) for each type.
2. Do this question on your own or as a class discussion. You have been given the following documentation task. Explain what existing documentation you might use to find the information you need.
   1. You are documenting a small database application you know that the user name fields allow only a certain number of characteristics to be entered. There is an error message when a user attempts to enter data into that field, so you need to look at documentation to identify the maximum allowed characters of the field.
   2. You are preparing an online tutorial for a popular small business accounts application. Bugs have been found in the code for some of the function keys so they are still in development, but you want to move ahead with your task anyway. You need documentation that will tell you the function keys’ commands.
   3. You are documenting a new network system, but are not sure what knowledge of networks you can expect most users to have. You want to make sure that you write your online manual at an appropriate level. You need documentation to tell you what group of users the system has been prepared for.
3. Referring to the user needs analysis you completed last week, identify the types of documentation currently available for program. Keep copies of the relevant sections of the documentation that explain how to use the features required by the user.

## Using the system to gain knowledge

You can also learn a lot about how the system works by using the system. This is a very simple and common way to learn about a system and offers some great advantages:

* It is direct. All types of documentation only offer second-hand knowledge about the system; using the system gives you first hand experience.
* It is accurate. On occasion, documentation does not accurately describe the finished system in every detail.
* It helps you to understand how the plans and designs have been put into practice.
* It puts you in the user’s shoes. By using the system in a way that your users do, you will have a clearer picture of how users perceive and work with the system.

On the other hand, learning from the system only, without looking carefully at the documentation first, can cause problems later.

* You may miss a feature or function. Only the documentation tells you what features and functions the system should have. If you do not know the documentation, you do knot know the features you need to document.
* A system may look or work slightly differently in different computing environments. For example, with different operating systems or types of hardware, a program might behave differently. Only documentation can tell you this is the case.

### Activity 2.2

Do this activity as a class or group discussion

Jan and Bob are having a lively discussion about the best way to learn about a system so that they can explain it to others.

Bob says “I think you always learn better by doing. Play with it, test it out and you’ll soon know what it can do!”

Jan says “I can see what you mean, but I think there is a bit more to it. You need to find out what the system does, and you learnt that best by looking at the specifications. How can you use the system effectively if you don’t know what the system does?”

What is your view and why?

## Analysing the system

When you have reviewed the existing documentation, you will have a good idea about what the system does and the functions and features it offers. The next step is to put what you have learned into practice – by using the system.

Examining the system is a process of looking at each screen, identifying the objects and fields on a screen, and asking: “What do they do?” and “What are they used for?”

For example, a button will perform a certain function if it is clicked. Or how do you move from one field to another? Or what happens when you right click on a particular area of the screen?

Any function that a user can access or use may need to be included in user documentation. You may also input deliberate errors in data, for example, to identify what error messages the system gives. This also identifies what needs to be included in user documentation. If a user receives an error message, they will need to refer to the documentation to determine the right data or information to provide.

## Using a systemic approach

You need to be organised and follow a plan in examining the system. Your approach might vary according to the type of system you are documenting, but there are some common features that you should examine carefully, including:

* Menus, both top and lower level
* Special menus such as the one used for drawing
* Buttons and/or shortcuts
* Icons

Common actions and functions include:

* Moving from one screen to another
* Moving from one field to another
* Closing the screen and the system

As you examine the various features, commands and functions, the purpose and use of the various screens should become clear. You should be able to see how screens are linked and related to each other. Other assistance may already be included with the system to help you understand it.

For example, many systems include message bars or mouse over help that are programmed in before the documentation task is complete.

Labels and icons can also help, especially if standardised symbols have been used, such as icons that match those commonly seen in the graphical user interface (GUI) of other programs.

When examining a system, it can be helpful to crate a checklist of the system’s features, commands and functions as outlined in the documentation. You can then check that you have examined each feature and function, and can make comments as required. Naturally, every system will generate a different checklist.

To write effective documentation, you need to understand the system. This cannot be done by examining the documentation on its own, nor by examining the system on its own. You need to use the information you find in the documentation and that you discover by running the system in order to document a system effectively. In addition, it is important to examine an unfamiliar system in a systemic way.

### Activity 2.3

For the user documentation that you are preparing to write for a friend/colleague to help them complete certain tasks, explore the software program they are using in detail. Pay close attention to the available features and functions that relate to the task(s) they require assistance with. Create a checklist that will enable you to keep a record of the features and functions to be included in the documentation.

# Writing Effective User Documentation

## Writing user documentation

A lot of preparation has gone into getting to this point, providing the tools that we need to make the writing task as easy as possible. The approved templates have given a structure to work from. Using recognised standards (where applicable and available) has ensured that the structure is a workable one that will give reliable results. In addition to this, by examining the documentation and the system, you will now have a good knowledge of how the system functions. Without this preparation, the task of actually writing the documentation would be much harder.

Writing effective user documentation is an exercise in effective communication, so the principles and practices of written communication and business writing will apply. Any form of communication will be greatly improved if you apply the following principles.

* Know what you want to say before you say it!
* Know who you are saying it to!
* Know how to say it!

## Know what you want to say before you say it!

This means identifying the *purpose or goal* of your communication: *why are you communicating?* Knowing the purpose of the documentation can help you to develop the user documentation as clearly and concisely as possible; that is, to get to the point! A communication goal or purpose may be a simple statement that you keep nearby to refer to when needed.

For example, a communication purpose may be:

*This document aims to provide users with simple step-by-step instructions to complete some of the more common features of the system.*

## Know who you are saying it to!

You need to know your *audience*, or who you are communicating to; that is, the users. This is important, as this will effect what you say, how you say it and the language that you use.

For example, the words and phrases you use when you are chatting with your friends will be very different to those used when chatting with your manager.

When writing to suit your audience, consider the following:

* The personal qualities of the audience, For example, if you are sending a letter to a group of people over 55 years of age, is it wise to ask them to respond via email?
* Whether it is appropriate to use technical language or ‘jargon’ that is specific to your industry. Pay particular attention to any industry jargon or technical terms that you commonly use but clients and suppliers may not be familiar with. The quickest way to create confusion and to cause others to switch off when communicating is to throw in few acronyms and computer-speak terms along the way.

For example, you might be able to use the acronym GUI to your colleagues, but will your message be helped or hindered if you use this acronym when talking to your grandmother?

* The belief and assumptions of the users. This includes users’ past experience with user documentation, education, training and so on.

Learning about or understanding the receiver (audience) will help you to target and structure user documentation more clearly.

## Know how to say it!

Regardless of what you are writing, and for whom, you must apply the rules and principles of writing in plain English. Plain English is a style of writing that ensures that the intended meaning of a communication can be easilyunderstood *the first time that it is read by the audience.*

### Writing in plain English

Writing in plain English has 3 distinct advantages:

1. It increases *efficiency.* If the documentation is easy to read and understand, the user will spend less time trying to comprehend the message.
2. It increases *equity.* All users will be able to understand the documentation, thereby increasing access and decreasing potential alienation.
3. It increases *effectiveness.* The user will immediately know what action they need to take to complete a task, as the documentation is clear.

For your user documentation (and indeed, any business writing) to conform to the principles of plain English, it must be *clear, concise* and *correct.*

### Making it clear

For user documentation to be clear, the reader must be able to understand the message immediately. If you convey more than one idea at a time, for example, you make it difficult for the reader to understand what you are really trying to say. A clear message is an effective message; it gets the point across immediately, and the reader knows exactly what action needs to be taken.

For example, the message you write may be short, perhaps only mouse-over messages, but they still need to be written clearly. A message may be short but still unclear. For instance, if you leave a floppy disk in the A: drive of a machine running a popular operating system, you will get a message ‘NLSLDR.NRM not loaded’, together with a blank screen. This message is certainly short but not clear to most people.

To improve the clarity of your writing, you will need to consider the following:

* The structure of your message, such as the length of sentences and the order in which information is presented. When writing, you should aim to present no more than one idea or point per paragraph.
* The layout of your documentation, and ensuring that it is presented in such a way as to attract the audience, as well as to enable them to easily identify the parts of the message that directly affects them. When you send a letter to a client, for example, they will most likely only read the second paragraph of a letter, which often contains the main message.

### Making it concise

Being concise means saying all that needs to be said, and no more! To be concise when writing effective user documentation means the following:

* Use simple language and avoid technical language where possible. Using four-syllable words might impress your manager, but it generally takes more energy for the reader to absorb big words as well as the overall message. Using language that the reader does not understand creates a barrier to communication, and reduces the chance of the message being understood or acted on
* Remove unnecessary words where possible. Whenever a reader receives a written communication, they will quite often scan to find the information that they need and ignore the rest of the message. If you use too many words in your communication, you risk frustrating and boring your reader. Consider the following plain English alternatives.

|  |  |
| --- | --- |
| **Example** | **Concise Alternative** |
| In other words, | So, |
| In conclusion, | Therefore, |

### Making it correct

The accuracy of user documentation is also important. If you ‘get the facts wrong’, not only have you conveyed to others that you are ill-prepared, but others may not refer to the user documentation in the future when needed. If this occurs, then all of your hard work up to this point has been in vain. To ensure that you provide the correct information, you must use the active listening techniques. Active listening ensures that you gather, convey and respond to the right information.

It is also important to check spelling, punctuation and grammar before the final draft is sent out. Errors can affect the credibility of the communicator and the organisation.

### Activity 2.4

To learn more about plain English and how it can help you write more effectively, go to <http://www.plainenglish.co.uk/>

For the user documentation you are preparing to write for your friend/colleague, describe each of the following:

* The purpose of the user documentation. Write a simple statement of purpose.
* The intended audience for the communication. Describe some of the characteristics of the user that may affect the way you write the documentation.

## Sequencing the documentation

Writing substantial documentation, such as a manual, can be a large task, but there are ways to help you stay organised. If you use a check list whilst you are examining the system, it can be a starting point for witting manuals it will list each function and each feature you need to write about. There are a number of approaches to choosing what order to write. Two common approaches are as follows:

1. Follow the functions. Describe each function that the system offers, one at a time, until you have described all the functions. A description of the function might then be followed by step-by-step instructions for how to use that function.
2. Follow the screens. In a smaller system, it is often efficient to describe each screen and describe the function that can be accessed from that screen. Step-by-step instructions can also be included to explain how to use the functions within a screen, with a brief description of that purpose of the function.

In the end, a mixture of approaches might be needed, especially if the system to be documented is large or complex.

User documentation usually provides a brief overview of the function and its purpose. This is then followed by step-by-step instructions for that function of feature.

For example:

**Adding shortcuts to the desktop**

You can add more shortcuts to the desktop that point to commonly used programs, folders or files.

1. In a blank area of the desktop, right-click your mouse to display a shortcut menu.
2. Click on **New** to reveal a submenu.
3. Click on **Shortcut** to display the **Create Shortcut** dialog box.
4. Click on **Browse**.
5. Navigate to the location of the file/folder that is to become a desktop shortcut.
6. Click on **OK**.
7. Click on the Next button to continue.
8. Type a name for the shortcut in the **Type a name for this shortcut:** field.
9. Click on **Finish** to close the dialog box.
10. The shortcut now appears on the desktop.

### Activity 2.5

For the user documentation you are preparing to write for your friend/colleague, create user documentation that includes three functions within the program for which your friend/college requires instructions:

1. Insert a heading to identify the function being described.
2. Provide a description of the purpose of that function.
3. Create step-by-step instructions for how to use each function.
4. Present the documentation in a logical order, that is, ensures that each function being described is presented in the documentation as it would follow in the program.

Exchange your work with another person or group (or submit to your teacher/assessor) to see how well others can use your instructions.